mechanics tutorial

Updated 16th October, 2014

1 Parameters

This section describes the model parameters that the user can access through the `parameters.h` file in each application folder. These can be divided between model-specific parameters and those generic to all models, and there are three categories of model-specific parameters: constant coefficients and tensors, bulk free energy functions, and residuals. In the context of their model, constant coefficients are self-explanatory. Residuals are used directly in evolution. Changing them is not recommended unless the user is comfortable with deal.ii and the PRISMS framework.

1.1 Generic Parameters

- `problemDIM`
  Dimension of the problem (e.g. 1D, 2D, 3D)
- `spanX`
  Length of system in x-direction
- `spanY`
  Length of system in y-direction. Not used if `problemDIM` < 2
- `spanZ`
  Length of system in z-direction. Not used if `problemDIM` < 3
- `refineFactor`
  Defines the refinement of the mesh. There are $2^{\text{refineFactor}}$ elements in each direction in this implementation, and $\left(2^{\text{refineFactor}}\right)^{\text{problemDIM}}$ elements in total.
- `finiteElementDegree`
  The order of interpolation of the finite element space. In this case, the order of the Lagrange elements to be used.
- `dt`
  The simulation timestep.

- `numIncrements`
  The number of simulation iterations. Final time is then `dt`·`numIncrements`.
- `writeOutput`
  Whether we are writing any output. Takes a boolean argument, e.g. `true`.
- `skipOutputSteps`
  Output will be written every `skipOutputSteps` iterations. If `writeOutput` `true`, the initial conditions will always be written.

1.2 `mechanics` Parameters

These parameters focus on building the stiffness tensor C_{ijkl} (`CijklV`), starting from a Young's Modulus (`Ev`) and Poisson's ratio (`nuV`).

$$\mu = \frac{E}{2(1+\nu)}$$

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$$

$$C_{ijkl} = \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \lambda\delta_{ij}\delta_{kl}$$

2 Boundary Conditions

Boundary conditions are all boundaries fixed ($u = 0$). Implementation of alternate boundary conditions is an objective for future releases.

3 Usage

```
$ make CMakeLists.txt
$ make
For serial runs:
$ make run
For parallel runs:
$ mpiexec -np nprocs ./main
```